Big Data Application Performance Monitoring in Retail E-Commerce using Spark

[1]Lavanya Marasa, [2]Kalyani Kunchum
[1]*Research Scholar, Department of Computer Science, ALTS, JNTUA University, A.P, India*
[2]*Assistant Professor, Department of Computer Science, ALTS, JNTUA University, A.P, India*

Abstract:

The global economy, today, is an increasingly complex environment with dynamic needs. Retailers are facing fierce competition and clients have become more demanding - they expect business processes to be faster, quality of the offerings to be superior and priced lower. Consequently, the quantum of data accumulate is at an all-time high as retailers generate giant volumes of data from numerous customer touch points across channels. For any fruitful business, we need to know more about customer preferences, interests, intent to purchase and more. It's important to have answers to questions such as: "who are my customers?", "what are they looking at?", "how similar are they to one another" and "what else might they be interested in viewing?". Apache Spark, the trendy big data processing engine that offers faster solutions for any failures compared to Hadoop, can be effectively utilized in finding patterns of relevance useful for the common man from these sites.

Keywords *- Big Data Analytics; Retail Stream Analysis; Spark Streaming, Data Analysis, resource constraints, application bottlenecks, Globally Unique Identifier(GUI).*

I. INTRODUCTION

The Big Data from the enormous sources available now have gained serious attention from researchers in every field with every attempt to maximize the value of knowledge resulting from its processing and analysis. Retails sites, now a day growing network site, are one such source which contributes huge data which possess values beyond customer's interests. Retail users can express or share their opinions, feelings or information regarding products, when each Customer starts browsing the shopping sites, for each transaction unique GUID (Globally Unique Identifier) will be generated and unique online id will be created and the online id will be stored as cookies on client machine for each customer. Handling such huge streaming knowledge victimization Spark system, that is taken into account because the second-generation huge processing engine, is that the topic of debate of this paper.

Retail sites often contain latest information of products as it is frequently updated. Even for every occasion the mega offers will be updated in the sites and send as alert messages, mails to registered customers. To enhance their business, they provide cashback offers so that customers will be interested to shop in those retail sites, due to this they will get good profits.

Even Retailer everywhere are using predictive analytics to determine which products to stock, the effectiveness of promotional events and which offers are most appropriate for consumers. Staples analyses consumer behaviour to provide a complete picture of their customers. predictive analytics to reduce risks, optimize their operations and increase revenue.

Across retail and client services, multiple trends have full stuffed growth in knowledge generation and can still propel the apace increasing pools of information. As data becomes an increasingly asset, shortening the lag time between generation and insight will be critical for companies to compete effectively. The use of information can become a key basis of competition across sectors, thus it's imperative that structure leaders begin to include knowledge management into their business plans—both from a cost-control viewpoint and a business-value viewpoint.

This paper investigates the problem of real time analysis is to monitor the performance of the applications and to detect if any failures to fix it without any delay to avoid business loss and filtering of those specific retail industries and predictive analysis of customer interests. The predictive analytics is really a game changer for Retail industries. A model needs to be proposed for the real-time collection and analysis of transactions for reach customers and the aggregation of transactions will be processed based on the browsing data of the customers in the retail industry

II. PREVIOUS WORK

Big Data processing requirements initiated a paradigm shift from traditional data processing, resulting in the evolvement of Map Reduce based frameworks like Hadoop. Though Hadoop has been extensively used for Big Data processing for years, performance wise a better solution like Apache Spark can be looked upon as a giant step in big data processing. The open source Apache spark ecosystem integrates batch and stream processing and comprise of libraries providing support for machine learning, graph processing and SQL querying.

Apache spark, originated from Berkeley, now licensed under Apache foundation offers much faster performance and a variety of features in comparison with the most sought out Hadoop Big Data Processing System. Though Hadoop is a matured batch processing system with many projects being completed and much expertise being available, it has its limitations. Hadoop is written in java and mainly rely on two functions, the Map and the Reduce, all operations are to be represented in terms of these two functions which makes the programming a little complicated. Spark program can be written using Java, Python or Scala and it offers more functions other than just the map and reduce and above all it provides an interactive mode, the spark shell, which makes programming much simpler for Spark compared to Hadoop. Hadoop persists data back to the hard disk after a map or reduce operation, while spark performs in-memory data processing and hence repetitive operations on same data will be done much faster. Hence memory requirement of Spark is higher compared to Hadoop but if the data fits in the memory, spark works faster or else it has to move data back and forth the disk which deteriorates spark's performance. Being a batch processing system, Hadoop users have to depend on other platforms like Storm for real time data processing, Mahout for machine learning or Graph for graph processing. But Spark ecosystem includes Spark streaming, MLib, GraphX and Spark SQL for real time data processing, machine learning, graph processing and SQL querying respectively, which gives competitive advantage for Spark over.

Spark application will be having a driver program that runs the main function and performs parallel operations on various nodes in a spark cluster.

By introducing the concept of Resilient Distributed Dataset (RDD) , the collection of immutable objects partitioned across the nodes of a cluster for performing parallel operations which can be persisted in memory for repetitive/iterative use, Spark outperforms Hadoop with100 times faster performance by saving time of read/write from disk, especially in running machine learning applications where iterative operations on data is common. RDDs are formed by transformation from other RDDs or file and RDDs retain the information by which it is formed.

Since Big Data analytics involves application of machine learning/data mining techniques on Big Data, Spark offers MLib, the machine learning library that includes popular machine learning algorithms for classification, clustering and association. Integration of MLib in spark ecosystem is another advantage that spark is having while Hadoop struggles with Mahout, the machine learning platform.

TABLE I
COMPARISON OF SPARK AGAINST HADOOP

Spark	Hadoop
Second generation Big Data processing engine, with extended features.	First Generation Big Data processing engine, matured, With much expertise available.
Availability of functions other than the Map and the Reduce, the option to write program in java, python or Scala and provision of interactive mode - the spark-shell makes programming easy.	Rely on just the Map and the Reduce functions, which makes programming difficult
Up to 100 times faster to Hadoop, especially in iterative operations, as intermediate data/result is persisted in memory	Slower as intermediate data/result is stored in hard disk
Spark being A batch processing Engine also includes spark streaming for streaming data processing, MLib for machine learning, GraphX for graph processing and spark SQL for querying thus providing an all-in-one solution.	Mainly A batch processing engine where users can depend on other compatible platforms for performing stream processing, machine learning or database querying.
Compatible with Hadoop Distributed File System(HDFS)	
Memory requirement is higher. Degradation in performance if data not fit in the memory	Lesser memory requirement

Spark streaming facilitates stream data processing, though spark is basically a batch processing engine. Incoming data stream is grouped into batches of interval less than a second and processed by the batch processing spark engine integrating the powerful features to near real-time processing. In this paper, we discuss the usage of Spark engine with MLlib, Streaming and Spark SQL for processing, classification and retrieval of career information

III. SPARK OVERVIEW

A. *Real time data Collectionand Processing using Spark Streaming*

Apache Spark provides programmers with an application programming interface centred on a data structure called the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way.[2] It was developed in response to limitations in the MapReduce cluster computing paradigm, which forces a particular linear dataflow structure on distributed programs: MapReduce programs read input data from disk, map a function across the data, reduce the results of the map, and store reduction results on disk. Spark's RDDs function as a working set for distributed programs that offers a (deliberately) restricted form of distributed shared memory.

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD transformations on those mini-batches of data. In figure 2 the design enables the same set of application code written for batch analytics to be used in streaming analytics.

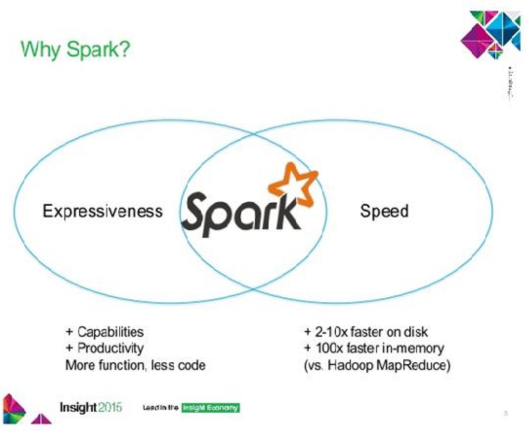

Figure 1: Spark Overview

After authentication, the streaming applications receives the retail stream and group them into batches with a suitable selection of batch interval. transactions are filtered at real time from the streaming based on browsing history. Using windowing function, all the relevant advertisements collected over a chosen interval of time is written to a text file. This intermediate result with listed transactions itself can serve as a source of information to the retailers.

B. Searching with Spark SQL

Transactions classified under various categories will be stored in the database, which can be queried to find vacancies belonging to a particular product category. Spark SQL provides querying functionality. Customer can query about a category and the advertisements regarding that category can be given as the result, spark SQL queries can be aggregated and can be performed various operations on SQLqueries based on the user requirement.

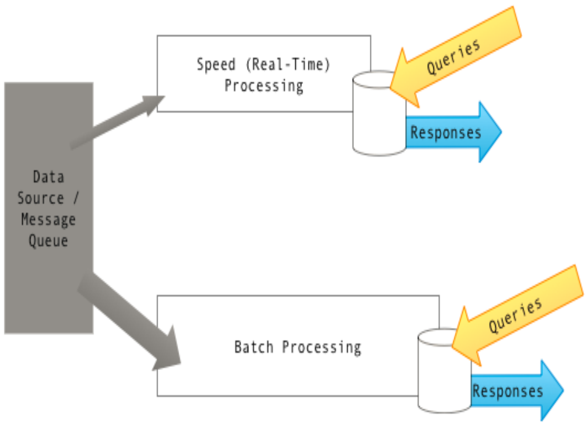

Figure 2: Lamba architecture

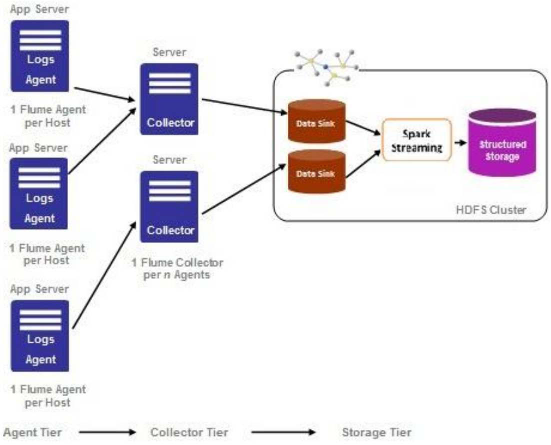

Figure 3: Application flow in Retail E-Commerce

IV. RESULTS

Spark application was run on a cluster with one master and two slaves, Streaming transactions are collected in batch interval of seconds after an initial filtering using time constraint. All transactions are collected and sent to Kafka clusters from their the data is read by the spark where the data will be aggregated based on the transactions and all the metrics like execution time, transaction count and CPU memory will be stored in terms of metrics calculations and the processed data is stored in Spark SQL where the data will be shown on the service layer API in terms of JSON format.

From the experiment, transactions are aggregate based on the time constraint and application and Globally Unique Identifier, and the execution transaction count is shown for aggregated transaction on service layer, internally the data read and write will be done by spark with high speed. Thereal time data collected based on the time constraints can be represented on dashboard to monitor the application performance to detect the application bottleneck.

V. CONCLUSION

In this Big Data era, Retail sites like Flipkart,Amazon are profoundly used for information regarding real time transactions analysis. This research work succeeded in developing and implementing a scalable model for real time analysis and filtering of product related transactions from among millions of click stream transactions and classify them into different categories that can lead to improving the business skills. In this work, Spark Streaming was utilized for handling the streaming transactions. Spark being open source and highly scalable, to identify the application performance and it can easily cater the needs of ever growing data size as well.

FUTURE WORK

The model explained in this paper can be generalized to serve variety of streaming applications. It can be extended to provide as a web service which sends near real time transaction details to registered users at fixed intervals of time. Also, the classified data from the present work can be further mined to find their customers interest, for different categories and different locations, at different periods of time in a year, as required.

REFERENCES

[1] Amarbir Singh and Palwinder Singh "Analysis of various Tools in Big Data Scenario", ISSN:2394-2231, vol. 03, Issue 02,Mar - Apr, 2016.

[2] Kiejin Park and Limei Peng "Second- Generation Big Data Systems," IEEE Computer, vol. 11, no. 14, pp. 8221-8225, 2016.

[3] S. Liu et al., "TASC: Topic-Adaptive Sentiment Classification on Dynamic transaction," IEEE Transactions on Knowledge and Data Engineering, vol. 27, no. 6, pp. 1696 - 1709,2015.

[4] T. Sakaki, O. Makoto and M. Yutaka, "Tweet analysis for real-time event detection and earthquake reporting system development," vol. 25, no. 4, pp. 919-931, 2013.

[5] AlexandarShkapsky, Mohan Yang and Matteo Interlandi,"Big Data Analytics with Datalog Queries on Spark",2016.

[6] Sparks, Evan; Talwalkar, Ameet (2013-08-06). "Spark Meetup: MLbase, Distributed Machine Learning with Spark". slideshare.net. Spark User Meetup, San Francisco, California. Retrieved 10 February 2014.

[7] Jump up ^ "MLlib | Apache Spark". spark.apache.org. Retrieved 2016-01-18.

[8] Malak, Michael (1 July 2016). Spark GraphX in Action. Manning. p. 9. ISBN 9781617292521. Giraph is limited to slow Hadoop Map/Reduce